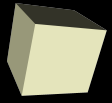




Binary Trees

3/29/2006





Opening Discussion

- Do you have any questions about the assignment?
- Do you have any questions about the reading?





Sorted Binary Trees

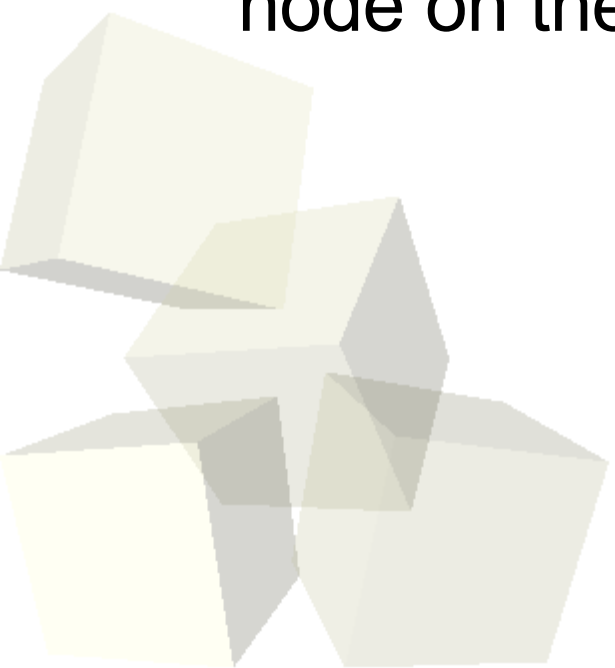
- One of the best uses of binary trees is the sorted binary tree. In this type of tree, we store data in every node and below any node we put lesser values to the left and greater values to the right.
- We find elements by going down the tree always going left or right. This gives us behavior like a binary search, but the tree is more flexible because adds and removes are quite efficient as well.





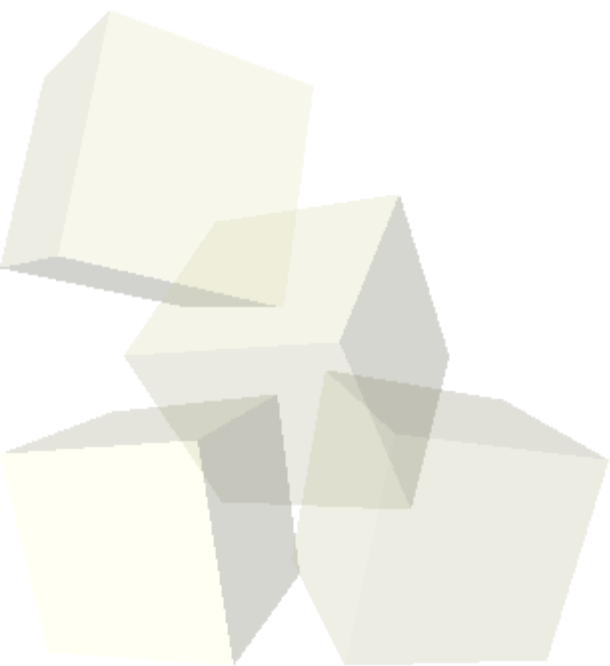
Adding and Removing

- The code for both adding and removing from a binary tree begins like a search that keeps track of previous (much like a linked list).
 - The add always goes to a leaf and adds the new element to the proper side.
 - The remove replaces the node we are removing with either the greatest node on the left or the smallest node on the right.





- I want us to code a BST together. We'll have one of you code and I'll walk through the process. These are really neat data structures so you really should get a feel for how they work.





- While support of threading was built into the original version of Java, some common threading tasks still aren't easy.
- There are some things that you find yourself doing rather frequently with threaded code and there are some limitations with the basic libraries that make it difficult to use at times.
- The `java.util.concurrent` library was added to make common parallel tasks easier. It is built around a few key concepts. Let's go look at the javadocs for this library.



- What can go wrong with the type of binary tree that we wrote today to make it perform poorly?
- Assignment #5 is due today unless you do the mock competition.

